

207626

ARK: AUTONOMOUS MOBILE ROBOT IN AN INDUSTRIAL ENVIRONMENT

S.B. Nickerson³, P. Jasiobedzki¹, M. Jenkin², A. Jepson¹, E. Milios², B. Down¹, J. R. R. Service³,
D. Terzopoulos¹, J. Tsotsos¹, D. Wilkes³, N. Bains⁴, T. Campbell⁴

N94-30529

¹ Dept. of Computer Science, University of Toronto, Canada M5S 1A4

² Dept. of Computer Science, York University, North York, Canada M3J 1P3

³ Ontario Hydro Technologies, Toronto, Canada

⁴ Atomic Energy of Canada Ltd., Mississauga, Canada L5K 1B2

Abstract

This paper describes research on the ARK (Autonomous Mobile Robot in a Known Environment) project. The technical objective of the project is to build a robot that can navigate in a complex industrial environment using maps with permanent structures. The environment is not altered in any way by adding easily identifiable beacons and the robot relies on naturally occurring objects to use as visual landmarks for navigation. The robot is equipped with various sensors that can detect unmapped obstacles, landmarks and objects. In this paper we describe the robot's industrial environment, its architecture, a novel combined range and vision sensor and our recent results in controlling the robot, in the real-time detection of objects using their colour and in the processing of the robot's range and vision sensor data for navigation.

1. Introduction

The ARK (Autonomous Robot for a Known Environment) Project is a precompetitive research project involving Ontario Hydro, the University of Toronto, York University, Atomic Energy of Canada Ltd., and the National Research Council of Canada. The project started in September 1991 and will be completed in August 1995. The technical objective of the project is to develop a sensor-based mobile robot that can autonomously navigate in a known industrial environment.

There are many types of industrial operations and environments for which the mobile robots can be used to reduce human exposure hazards, or increase productivity. Examples include inspection for spills, leaks, or other unusual events in large industrial facilities, materials handling in computer integrated manufacturing environments, and the carrying out of inspections, the cleaning up of spills, or the carrying out of repairs in the radioactive areas of nuclear plants - leading to increased safety by reducing the radioactive dose to workers.

The industrial environment is significantly different from office environments in which most other mobile ro-

bots operate. The ARK project will produce a self-contained mobile robot with sensor-based navigation capabilities specifically designed for operation in a real industrial setting. The ARK robot will be tested in the large engineering laboratory at AECL CANDU in Mississauga, Ontario (figure 1). This open area covers approximately



Figure 1. A view of the AECL industrial bay

50,000 sq. feet of space and accommodates one hundred and fifty employees. Within the Laboratory, there are test rigs of various sizes, mock-ups of reactor components, a machine shop, a fabrication facility, metrology lab and assembly area. There are no major barriers between these facilities and therefore at any one time there may be up to fifty people working on the lab floor, three fork lift trucks and floor cleaning machines in operation. Such an environment presents many difficulties that include: the lack of vertical flat walls; large open spaces (the main isle is 400' long) as well as small cramped spaces; high ceilings (50'); large windows near the ceiling resulting in time dependant and weather dependant lighting conditions, a large variation in light intensity, also highlights and glare; many temporary and semi-permanent structures; many (some very large) metallic structures; people and forklifts moving about; oil and water spills on the floor; floor drains (which could be uncovered); hoses and piping on the floor; chains hanging down from above, protruding

structures, and other transient obstacles to the safe motion of the robot ¹¹.

Large distances, often encountered in the industrial environment, require sensors that can operate at such ranges. The number of visual features (lines, corners and regions) is very high and techniques for focusing attention on specific, task dependent, features are required. Most mobile robotic projects assume the existence of a flat ground plane over which the robot is to navigate. In the industrial environment this ground plane is generally flat, but regions of the floor are marked with drainage ditches, pipes – this requires sensors that can reliably detect such obstacles.

The ARK robot's onboard sensor system consists of sonars and one or more ARK robotic heads and a floor anomaly detector (FAD). The head consists of a colour camera and a spot laser range finder mounted on a pan-tilt unit ⁵ (see also figure 3). The pan, tilt, camera zoom, camera focus and laser distance reading of the ARK robotic head are computer controlled. The ARK project is investigating different technologies for Floor Anomaly Detection (FAD) to detect objects on the floor that cannot be detected by the sonar system and are too large for ARK to traverse. One technology that is being developed is a laser based system built around the NRC BIRIS laser head¹. A second approach is to use stereo vision to localize potential floor anomalies. Unlike the classical approach to stereo, the stereo based FAD uses calibrated non-zero torsional eye positions to warp the disparity surface to simplify the process of detecting structures near the ground plane ⁹.

The ARK robot navigates in its environment without help from a human operator and with no engineering of the environment through the addition of radio beacons or magnetic strips beneath the floors. Also, modification of the environment to include unique and easily identifiable beacons is also not permitted. The robot uses naturally occurring objects as landmarks. The robot relies on vision as its main sensor for global navigation, using a map with permanent structures in the environment (walls, pillars) to plan its path. While executing the planned path, the robot searches the environment for known landmarks. Positions and salient descriptions of the landmarks are known in advance and are stored in the map. The robot uses the relative position of the detected landmark to update its position. The robot's visual tasks include detection of landmarks and searching for known objects. The robot avoids any objects in its path by using the reactive part of its control system. These objects

could be stationary or moving, and do not have to be a part of the internal representation.

In this paper we describe some recent research aspects of the project. In particular we concentrate on environmental path planning, the reactive control system, colour based detection of objects and 3D scene segmentation using the combined visual / range sensor.

2. Mobile Platform and Sensors

We are building two ARK prototypes: one at the University of Toronto and the other at AECL. ARK-1 (at Toronto) is being jointly constructed by university and industry personnel. We use ARK-1 to test the ideas, sensors and algorithms that will ultimately be included in ARK-2. The computing for ARK-1 is done mainly off-board while that for ARK-2 will be done mainly on-board. Both robots use visual data obtained through active vision processes as a primary source of sensing for the robot. They also use non-visual sensors such as infrared, sonar and laser range-finders. Both ARK robots use the Cybermotion Navmaster platform as their mobile base (see figure 2).



Figure 2. The ARK-1 robot

2.1. Mobile Platform

The main hardware components of the ARK-1 robot are: the Navmaster mobile platform from Cybermotion, the robotic head with sensors and a remote link to a host computer network (figure 2). The platform consists of a base with three wheels and a rotating turret. A bumper, equipped with contact sensors, is mounted to the turret. The turret was originally equipped with six sonars: two of them face forward, two backward and two sideways. Each sonar emits a cone shaped acoustic wave and can

detect the reflected wave. The time required by the sound to travel from the robot to an object and back gives a measure of the object distance. We have experimented with using additional sonars mounted on the turret or the bumper to enhance the interpretation of the sonar data.¹⁴ Multiple return signals were combined in a three dimensional grid in robot coordinates using a Bayesian update rule. Additional readings were obtained by small movements (less than 1 m) of the robot. This approach helped to map more accurately obstacles in front of the robot and to reduce the influence of noisy return signals.

The ARK-1 robot communicates with a network of host computers via the 8-channel remote serial link. The communication between the robot and the host is on the level of processed signals from sensors and commands sent to the robot. The on-board computers collect the data from various sensors, preprocess it and send it via the radio link to the host computer network. The computers in the network analyse this data, and generate commands for individual units of the robot (platform, head, sonar controllers, range-finder). The on board computers perform time critical functions such as emergency stop, positioning the head and moving the platform. The host network of computers consists of a multiprocessor SGI Power Series 4D380 and several Sun SPARC 2 workstations, all running under the Unix operating system.

In ARK-2, most of the computation, such as processing and interpretation of data from various sensors and generation of control commands, will be done on board. The communication link will be primarily used for exchanging messages between the robot and the operator. The on board computer will operate under control of a real time operating system.

2.2. Combined Vision / Range Sensor

We have installed a special sensor (Laser Eye) on the ARK turret. This sensor can provide colour images and range data at distances up to 100 m which are typical for the industrial environment. The Laser Eye is a combined range / video sensor consisting of a camera and a laser range-finder⁵. The range-finder uses the time-of-flight principle and provides a single depth measurement for each orientation of the sensor. Measuring distances to objects in the scene requires pointing the sensor at each of them in turn and reading their depth. The range-finder uses an infra-red laser diode to generate a sequence of optical pulses that are reflected from a target. The time required to travel to and from the target is measured to estimate the distance. The laser is eye safe – this permits its use in the presence of people.

Our robotic head has four degrees of freedom: two extrinsic – head pan and tilt, and two intrinsic – camera zoom and focus (figure 3). The head can tilt in any direc-



Figure 3. The robotic head with a combined visual & range sensor (Laser Eye)

tion between 65 degrees below and 95 degrees above the horizon and the panning range covers 360 degrees. The head can rotate with speeds exceeding 180 degrees per second. Figure 3 shows the first model of the head with the Laser Eye sensor.

The range-finder measures distance to an object in the centre of the camera field of view. The co-linearity of the camera optical axis of and that of the range-finder is achieved by using a hot mirror (one that reflects infra-red and transmits visible light) placed in front of the camera lens. The mirror transmits the visible light from the observed scene to the camera with minimum attenuation. The hot mirror reflects the transmitted infra-red beam and sends it in the direction of the optical axis of the camera. The returning pulse is reflected by the hot mirror again and projected on a detector in the range-finder⁵. A single range measurement takes 0.12 – 0.5 second depending on the selected accuracy. The time required to

point the head in a new direction depends on the required rotation.

3. Control Architecture

The ARK control system consists of two levels: a high level and a low level reactive system. The high level is responsible for planning robot actions, global path planning, selecting landmarks for sighting and interactions with the user. The low level, reactive component of the control system, uses the on board obstacle avoidance system of the platform to detect obstacles and to navigate around them.

The path planner assumes that the low level reactive control structure will safely execute segments of the plan in the presence of unmodelled or unexpected obstacles. By breaking the path planning process into a GOFAIR (Good Old Fashioned AI and Robotics) task which can be processed using classical AI tools, and a real time reactive process which can be processed using a real time safety critical system implemented as a subsumption architecture, ARK takes advantage of the best of both paradigms.

3.1. Position Estimation and Global Path Planning

The global navigation system uses visual landmarks to update the robot position estimate. A dead reckoning system on the platform measures the distance travelled and provides the current orientation. The positional error introduced by the dead reckoning system accumulates over time and has to be reset by measuring the robot position with respect to landmarks stored in the map. The map is represented as a 2D floor plan that contains permanent objects, semi-permanent objects entered by the user, obstacles detected by the robot and landmarks. Each location in the map is annotated with landmarks that are visible from this location. We use a Kalman filter to update the current position estimate ⁸.

The global path planning process represents the world as a two dimensional grid. We have experimented with various path planning algorithms such as the shortest path, the minimum cost, and the minimum uncertainty. The shortest path minimizes the distance travelled by the robot and the minimum cost minimizes the number of grid cells visited by the robot. The minimum uncertainty path planner uses the known position of landmarks to choose paths that minimize the expected uncertainty from the start position to the goal. By selecting such a path, the robot may travel a longer distance but its positional error along the path will be much smaller as it can update its position estimate more often.

Figure 4 shows a user interface displaying a map, robot and a planned path. The interface facilitates the creation of a map of the environment, as well as the planning and execution of a path by the real or simulated robot. The high level control system assumes the presence of a low level reactive control system that can execute the path created by the high level.

3.2 Reactive Control

The high level planner communicates with the reactive subsystem through a very simple set of operations that assumes the reactive phase of the planner will operate autonomously and asynchronously; attempting to achieve the current subgoal ¹². The low level control of the robot is based around the subsumption approach described by Brooks ².

The robot is guided by a set of behaviours that operate in parallel. Each behaviour maps a sensory reading from the robot's environment into an external action of the robot. Conflicting behaviours are arbitrated based on an absolute prioritisation of behaviours. There are three basic behaviours that control the robot: move, avoid, and escape. Avoid watches for an obstacle detected by the front sensing sonar. If an object appears the avoid behaviour stops the robot, and turns it to a new direction so that the robot will not collide with the obstacle. The escape behaviour watches for an obstacle directly in front of the robot, in which case, it causes the robot to back-up and then to turn to a new direction. The escape behaviour helps to get out of certain deadlocks that may occur with the avoid behaviour when the robot gets stuck in a corner. The move behaviour steers the robot towards a precomputed goal position.

Figure 5 shows the planned path and the reactive path executed by the robot as it moves through a doorway. The robot starts in the right top position and moves until it approaches the doorway. At this point, the avoid behaviour is triggered by the edges of the doorway.

4. Using Vision for Navigation

Computer vision plays a major role in the ARK project. The ARK robot uses vision to detect and track landmarks and to search for other known objects. Subsequent surveys and preliminary vision testing have yielded many potential candidates for ARK landmarks in the AECL bay. It is important that these landmarks not only image well but that their occurrence be frequent. Typical landmarks within the AECL laboratory consist of alpha-numeric location signs, fire extinguisher markers, door-

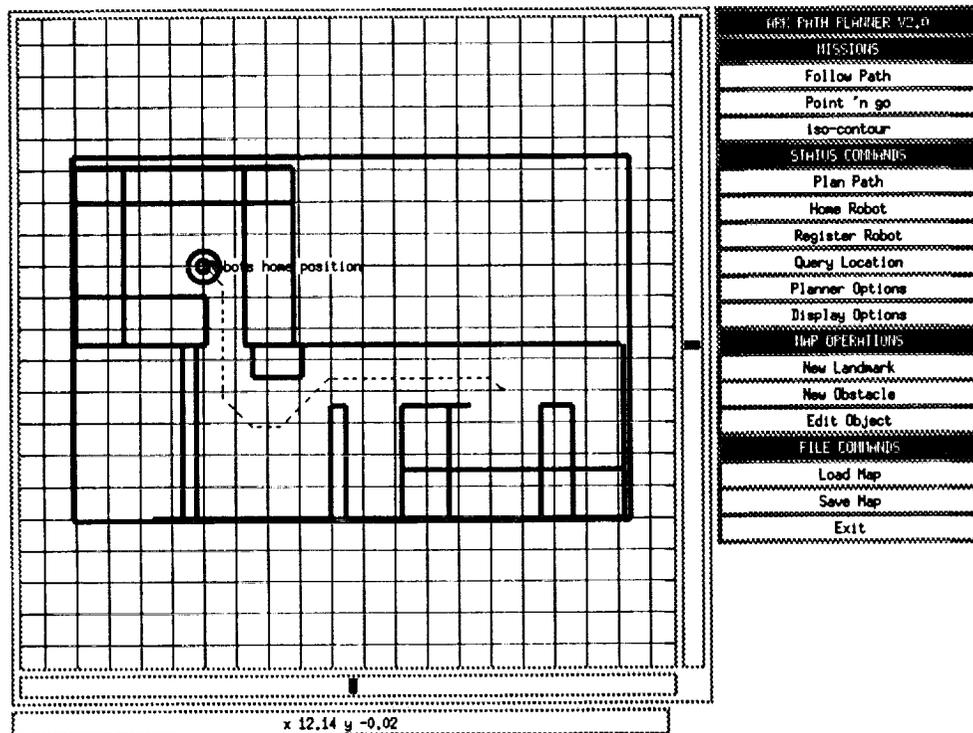


Figure 4. Path planner interface

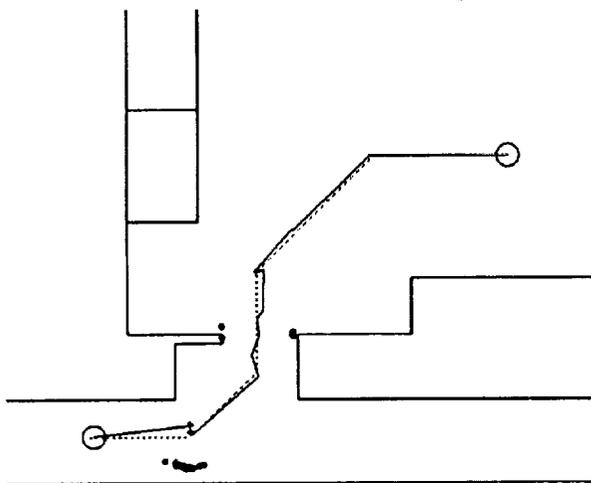


Figure 5. Planned and executed path

ways, overhead lights, and pillars. The only criteria used is that they are distinguishable from the background scene by colour or contrast. These criteria allow the use of both grey level and colour image processing algorithms for landmark identification.

Vision provides important information where to point the range-finder to obtain the most important information. This location depends on the current task, for example, detecting an obstacle or a passage between obstacles. It

also depends on the state of a data processing and is driven by an attention model. In two following sections we present results of using vision to detect objects using their colour and to select targets for range measurements.

5. Detecting Landmarks and Objects Using Colour

Visually searching for objects requires scanning the environment or checking expected locations with a camera or even moving a robot. In typical tasks of detecting visual landmarks or searching for a target object, the object itself and its salient characteristic is known in advance. When searching for a landmark the robot can predict where to point the camera as it knows its own approximate location on the map and the coordinates of the landmark. Still, uncertainty of the robot's position requires selecting a wide field of view for the camera. An attention mechanism that selects some "interesting" locations in an image or environment significantly speeds up and simplifies the search. Features such as intensity, colour, high contrast, motion and presence of significant edges are often used to focus attention. Once candidate locations have been selected, each of them is inspected closely to verify presence of the target object.

We use colour to identify possible candidates in an image. The colour classification scheme consists of an off-line training phase and an on-line classification of pixels on a real-time image processor⁷. Colour informa-

tion is used for pixelwise classification of images and assigning pixels to possible target candidates or background classes. We apply classical methods of pattern recognition for pixel classification. We achieve the real-time performance by creating look up tables (LUTs) during the training phase and fast indexing during the on-line classification.

5.1. Real-time Colour Classification

Classification of every pixel in the image is a computationally expensive task. Modern image processing systems are often equipped with large look up tables that allow for real-time processing of every pixel. Combination of multiple data streams, for example RGB, into one channel enables us to index into the LUT and achieve the real-time performance of an arbitrary (non-linear) conversion. The nature of this conversion is determined by the contents of the LUT. The problem is how to create a LUT that will effectively capture the important variability of the data.

Resolution of the feature space can reach 2^{24} (3×8 bit colour bands) for standard colour cameras. Often it is sufficient to operate on smaller arrays. There are hardware limitations as well, for example, the Datacube MV20 advanced processor, used in the project, has a look up table with a maximum of 64 k entries. The contents of look up tables are often determined by manual selection. A more systematic approach uses training by showing examples and manually delineating the objects of interest. Cells in colour space, corresponding to the feature combinations present in the training set, are assigned to appropriate classes. For low resolution of the feature space (200 cells) such a technique is sufficient, as camera noise and blur create dense clusters¹³. For high resolution look up tables containing, for example 64 k cells, this approach is not reliable as insufficient training data creates "holes" in the feature space. Such holes cause misclassification of the data. Various heuristic techniques of filling the space have been used to bridge the gaps¹⁰.

To overcome the problem of the gaps in the LUTs created by limited number of training combinations, we use classical statistical pattern recognition techniques to fill the table. The brute force classification of all possible feature combinations fills the LUT easily.

The training sets consist of images with objects of interest in their natural environment and under different illuminations. Each pixel in the training set is described by its three colour components (RGB or HSI depending on the selected colour space). A clustering programme parti-

tions the three dimensional feature space and creates descriptions for all clusters detected in the training set. After clustering the user assigns individual clusters to classes corresponding to the trained objects and the background. A classification programme uses the description of clusters and their class assignment to process all the pixels in a test image. The test image contains all the feature combinations for a given resolution of the feature space and the resulting LUT will have all its cells filled by this process. Resolution of the LUT is limited by the image processing hardware and in our case the LUT size is equal to 64k (2^{16}). Decomposition of the 24 bit input data into 16 bits can be constant and may always rely on the same algorithm. Alternatively, it may vary depending on the distribution of data in the feature space.

The on-line classification combines the colour components of every pixel into one index to address an entry in the look up table. This entry contains a label corresponding to one of the trained classes.

5.2. Implementation and Results

We have implemented the training phase (clustering and creation of the LUT) on a Unix host. The real-time colour classification is being implemented on the MaxVideo 20 image processing system.

We trained the classifier to detect red and green circular plates similar to the ones displayed on the wall in the scene shown in figure 6. The training set contained mul-

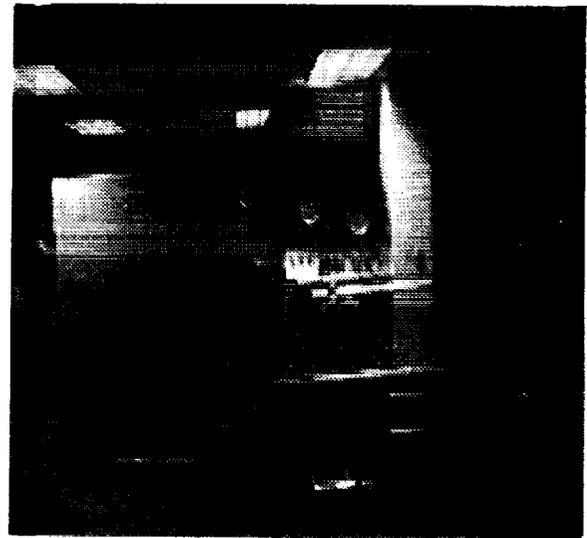


Figure 6. An office scene with coloured objects (luminance is shown only)

multiple plates located in various locations in the scene. The illumination varied between locations. The original pixel

values were represented in the RGB space. We used the K-means algorithm to group the data into approximately 20 clusters. The user assigned clusters corresponding to plates to three classes: red, green and the background. This technique is described in detail in 7.

Figure 7 shows the results of pixelwise classification,



Figure 7. real-time colour detection and reconstruction of object candidates from figure 6

filtering and reconstruction of large blobs representing red and green classes. The results of this processing are not perfect – both red plates have been detected but among the four green candidates only one corresponds to the target object. Also, detection of individual plates is not perfect as regions in the shade or reflecting light are misclassified. Different techniques could be used to decide whether the detected blobs correspond to valid objects or not. At this resolution, however, it might be difficult to decide if the shape deformations are caused by noise, particularly if the sensor is positioned at a difficult viewing angle. It is much better to point the robotic head at every candidate in turn and then acquire and process a new set of images.

Each detected candidate is described by a set of parameters that define its position in the image, size and location of its bounding window. The new orientation of the head is calculated from a kinematic model of the head that includes the pan, tilt and the initial size of the field of view. The new setting for zoom is selected so that the blob of interest is fully included in the new view but dominates the field of view.

6. Using Vision and Range for Navigation

The robotic head with the Laser Eye provides colour images and sparse range measurements at distances up to 100 m. With the current version of the head we can obtain sparse range measurements at a rate over 2 Hz. For the real-time operation of the robot it is important to minimize the number of measurements. We use image data to plan where to point the range-finder 4, 5.

6.1. Region Based Image Representation

We assume that nearly all significant depth discontinuities in the scene coincide with the boundaries of detected regions. This assumption requires that the initial segmentation creates an over- rather than under-segmented representation of the image. The under-segmentation can cause potential problems as it requires additional depth measurements to split the region along a depth discontinuity. The size of the regions should not be too small as it is difficult to obtain reliable distance measurements for small regions due to the finite size of the laser spot and accuracy of the robotic head.

The initial segmentation creates an image tessellated into primary regions of homogeneous image properties (intensity, colour, etc.). The segmentation method adopted for the project consists of smoothing, morphological edge detection and the watershed transform. This has been described in detail elsewhere 4. Large numbers of closed regions of similar image properties are created as a result.

In the image of AECL bay, shown in the figure 1, depth varies from approximately 3 m to 100 m. Figure 8 shows regions detected in figure 1 by the segmentation algorithm. A range map corresponding to this scene can be

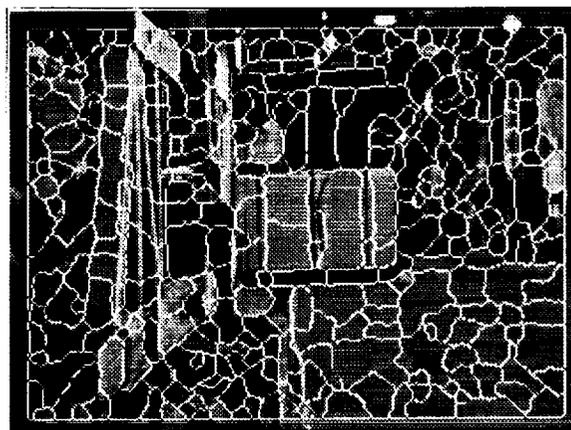


Figure 8. Image from figure 1 segmented into regions created by selecting target points for each region and

pointing the sensor at each of them. The number of targets required for each region depends on the world model and the required robustness. In a simple example, a single range measurement per region yields an approximate range map. Orientation of a planar surface in 3D can be recovered by measuring the distance to at least three points for each region and fitting a plane in Cartesian coordinates. Further processing uses the distances to targets and properties of regions and curves. The result of this processing is a 2 1/2 D representation of the scene.

6.2. Attention Driven Target Selection

In the example shown, the initial segmentation created almost two hundred primary regions. Assuming the simple model with one range measurement per region, creation of the complete range map requires almost 200 range measurements. By applying the above technique we have been able to reduce the number of range measurements required to create the dense range map from 64k samples (sampling every pixel in a 256x256 grid) to a much more manageable number of 200 to 1000 samples (200 regions x 1...5 targets per region). This has been achieved if the initial over-segmentation of the image identified intensity discontinuities and that they account for nearly all the depth discontinuities. For the mobile robot, operating in real-time, this may still be too slow. If we look at the intensity image ourselves, it seems that a few range measurements, taken at the "right" orientations, could provide the essential information essential for a specific task. We decided to look to models of human attention for inspiration.

The attention scheme, used here, depends on three components⁶:

- i. *a priori* information,
- ii. selection of salient features,
- iii. a given task and previous results of attentive processing.

The *a priori* information is encoded as a function biased to look at specific parts of the image. This function represents preferred behaviour (directional sensitivity) of the system, for example, data in the centre or below the horizon might be more important than at the periphery of the camera image.

Representing the segmented image data as a graph allows easy access to underlying regions and boundaries in the graph and for access to adjacent ones. The regions are described by features such as intensity, colour, texture descriptors, and their size and shape. The boundaries between adjacent regions are described by their size, shape,

orientation and contrast between regions on both sides. Detection of winners, in the Winner Take All scheme³, uses a combination of these features and is biased by the specific task performed by the robot.

For example, looking for a passage might involve searching for a dark region in the image. Depth discontinuities are likely to occur at boundaries between contrasting regions. If the task is to provide a qualitative range map, then selecting large regions first will enable faster coverage of the image by range data. Results of previous range measurements can influence the selection of the next target. This selection is task dependent. For example, when searching for an obstacle, if a depth discontinuity is detected, then the next ranging operations should concentrate on recovering the full extent of the closer object and not the distant one. If such a discontinuity is detected while searching for a passage then the successive ranging operations should concentrate on objects further away – the opposite strategy.

Figure 9 shows the attended receptive fields and the path of 10 saccadic movements between regions of high intensity. The initial bias is uniform and contributions from all



Figure 9. Bright regions selected by a uniformly biased attention model

receptive cells (pixels) are treated equally and, as the result, large bright regions are attended first. Edges of high contrast are likely locations for depth discontinuities. Boundaries between regions now serve as salient features. Pointing the range-finder at a boundary is not practical so two regions on both sides are selected for attention. Figure 10 shows a sequence of saccades between contrasting regions with a bias to the central part of the image. To minimise the number of measurements, each region is attended only once even if it is selected by two different boundaries.



Figure 10. High contrast regions selected by a centrally biased attention model

7. Discussion

The ARK robot relies on its combined vision and range sensor to navigate through the industrial environment. This sensor is unique as it operates at large distances that are typical for the industrial setting. Such distances are not covered by other available techniques used by mobile robots: stereo and active triangulation. Long distance sensory data allows the robot to detect landmarks, search for objects and possible paths well in advance. Early detection of such situations allows the robot to modify its trajectory or to change the plan without the need for an exhaustive search of the environment. Our work concentrates now on extending the reactive, subsumption based, control architecture by implementing additional behaviours. At present, we are moving now with our experiments from the university laboratories to large open spaces of the AECL industrial bay.

One of the strengths of the ARK project stems from the close working relationship between the industrial participants and the researchers from the University of Toronto, York University and the National Research Council.

8. Acknowledgements

Funding for this work was provided, in part, by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., the Department of Industry, Science and Technology Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited.

9. References

1. Blais F., Rioux M., Domey J.: "Optical Range Image Acquisition for the Navigation of a Mobile Robot".

Proc. of IEEE Int. Conf. on Robotics and Automation, 1991.

2. Brooks R.: "A Robust Layered Control System for a Mobile Robot". IEEE Trans. on Robotics and Automation, 2(1), 1986, pp. 14 - 23.

3. Culhane SM, Tsotsos JK: "An Attentional Prototype for Early Vision". ECCV-92, pp. 551 - 560.

4. Jasiobedzki P.: "Active Image Segmentation using a Camera and a Range-finder". Applications of Artificial Intelligence XI: Machine Vision & Robotics. Orlando, Florida, April 1993, p. 92 - 99.

5. Jasiobedzki P., Jenkin M., Milios E., Down B., Tsotsos J., Campbell T.: "Laser Eye - a new 3D sensor for active vision". Intelligent Robotics and Computer Vision: Sensor Fusion VI, Proc of SPIE, vol. 2059, Boston, Sept. 1993, pp. 316 - 321.

6. Jasiobedzki P., Service J.: "Recovering Depth by Saccadic Movements of an Active Ranging System". Conference on Vision and Pattern Recognition, CVPR 94 (submitted).

7. Jasiobedzki P., Down B., Service J. Wu V.: "Active object detection using colour and shape". 8-th Canadian Conference on Computer Vision, Signal and Image Processing, Vision Interface 94, Banff, May 1994 (submitted).

8. Jenkin M., Milios E., Jasiobedzki P., Bains N., Tran K.: "Global Navigation for ARK". Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS '93, Yokohama, Japan, July 26-30, 1993, pp. 2165-2171.

9. Jenkin M., Tsotsos J.: "Active Stereo Vision and Cyclo-torsion." Conference on Vision and Pattern Recognition, CVPR 94 (submitted).

10. Massen R., Volk G.: "Real-time colour classification for preprocessing photogrammetry images". SPIE vol. 1395, Close-Range Photogrammetry Meets Machine Vision, pp. 283 - 290.

11. Nickerson B., Jenkin M., Milios E., Down B., Jasiobedzki P., Tsotsos J., Bains N., Tran K.: "ARK - Autonomous Navigation of a Mobile Robot in a Known Environment." Proc. of International Conference on Intelligent Autonomous Systems: IAS-3, Pitsburgh, PA, February 1993, pp. 288 - 296.

12. Robinson M., Jenkin M.: "Reactive Low Level Control of the ARK". 8-th Canadian Conference on Computer Vision, Signal and Image Processing, Vision Interface 94, Banff, May 1994 (submitted).

13. Swain M., Ballard D.: "Color Indexing." IJCV 7:1, pp. 11-32.

14. Wilkes, D., Dudek, G., Jenkin, M., and Milios, E., "Multi-transducer sonar interpretation". IEEE Int. Conf. on Robotics and Automation, Atlanta, GA, 1993, vol. 2, pp. 392 - 397.